



Prévision du trafic à court terme par réseau de neurones

T. Dochy, Mehdi Danech-Pajouh, Yves Lechevallier

► To cite this version:

T. Dochy, Mehdi Danech-Pajouh, Yves Lechevallier. Prévision du trafic à court terme par réseau de neurones. 1994. hal-00542226

HAL Id: hal-00542226

<https://hal.science/hal-00542226>

Preprint submitted on 2 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Prévision du trafic à court terme par réseau de neurones

Thierry Dochy ¹, Mehdi Danech-Pajouh ¹, Yves Lechevallier ²

¹INRETS-DART

2 av du Général Malleret-Joinville
94114 Arcueil

²INRIA

Domaine de Voluceau
78153 Rocquencourt

La connaissance du futur proche en terme de trafic, permet aux exploitants d'améliorer et d'adapter le service offert aux usagers. C'est pourquoi toute recherche permettant d'améliorer cette connaissance mérite d'être entreprise. Partant de là, il nous est apparu intéressant de promouvoir une méthode de prévision à court terme issue d'une démarche neuronale.

L'objectif fixé est la réalisation d'un modèle de prévision à court terme fondé sur l'utilisation de réseaux de neurones à couches, calibrés heure par heure. Les potentialités de cette méthode sont évaluées par une application au trafic autoroutier pour laquelle on se fixe l'objectif d'un taux d'erreur inférieur à 10%.

Le problème de la prévision à court terme peut être résolu de plusieurs manières, notamment à l'aide de séries temporelles. Cette notion impose au niveau des informations une contrainte de continuité temporelle, c'est à dire une suite de mesures portant sur une période longue et continue. La prévision est alors réalisée à partir des processus AR, ARMA, ARIMA, dont les paramètres peuvent être estimés à l'aide de la méthode définie par Box et Jenkins [1970]. Cependant il n'est pas toujours possible de respecter cette contrainte de continuité en raison des pannes de capteurs ou d'indisponibilité des données.

A l'opposé des méthodes fondées sur les séries temporelles, il existe une approche dans laquelle les mesures portent sur des intervalles plus courts pas forcément continus. Dans ce contexte on peut citer la méthode ATHENA [Danech et Aron, 1990] qui combine les outils de classification et de régression. Cette méthode repose sur une analyse horaire du trafic, c'est à dire que pour chaque heure de la journée on estime les paramètres du modèle prévisionnel. Concrètement on extrait les données utiles à la prédiction de l'instant t , on calcule ensuite des données profils - c'est à dire rapportées au total des débits sur une période que l'on choisit - sur lesquelles on réalise une classification.

Pour chaque groupe on effectue un ajustement par une régression linéaire multiple. Quant à la prévision il suffit d'effectuer une reconnaissance de la classe d'appartenance, puis d'utiliser les coefficients de régression de cette classe sur les données disponibles.

Bien que l'application des réseaux de neurones au domaine prévisionnel reste très récente, on trouve toutefois quelques travaux dont la particularité est d'utiliser un seul réseau de neurones, à savoir :

- Prédiction de consommation journalière de courant à EDF [MANGEAS, 1993]
- Prédiction de trafic à l'Université de Leeds [DOUGHERTY 1993] et au LIAC[MONDE, 1991]

Quant à notre approche, on peut dire qu'elle repose sur la même hypothèse qu'ATHENA à savoir une forte dépendance entre le trafic et l'heure. Cependant à l'instar d'ATHENA elle nécessite moins d'arbitraires comme le nombre de classes, le nombre de coefficient en fonction des effectifs des classes.

A l'opposé des séries temporelles qui supposent la détermination des coefficients à partir de données contiguës, nous estimons les paramètres du réseau grâce à des mesures relevées sur des jours différents pour des instants communs. A l'aide des vingt-quatre réseaux de neurones notre méthode procure un ajustement horaire plus souple que celui proposé par les séries temporelles qui utilisent souvent les mêmes coefficients pour une journée.

Rappel sur les réseaux de neurones

Un réseau de neurones est caractérisé par plusieurs entités :

- **le neurone**, coeur du réseau; c'est une cellule reliée à des cellules voisines, qui émet un signal en fonction des impulsions reçues.
- **la topologie du réseau**, c'est la façon dont sont reliés les neurones ; ils peuvent être tous connectés ou partiellement connectés.
- **l'évolution du réseau**, on sous-entend la manière de modifier les liaisons inter-neurales ou les poids des connections.
- **le poids d'une connexion** pondère une liaison entre deux neurones.
- **la fonction de transfert** matérialise la sortie d'un neurone en fonction des impulsions reçues.

Soit i l'indice du neurone, on appelle e_i la somme pondérée des impulsions reçues :

$$e_i = \sum_{j=1}^n W_{ij} S_j$$

on appelle S_i la sortie du neurone i , cette sortie est fonction des impulsions reçues en

provenance des neurones j pondérées par les connections W_{ij} . Cette fonction est tracée en figure 1.

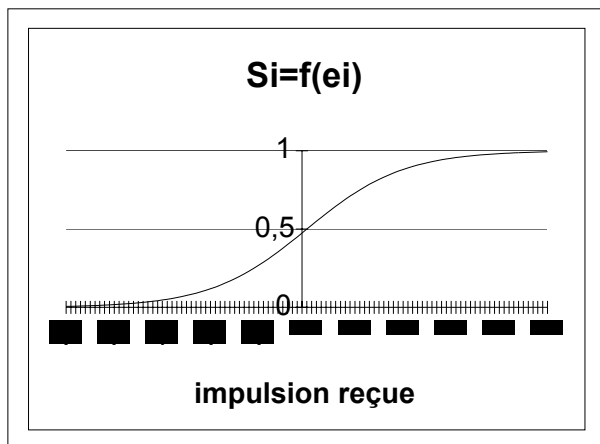


Figure 1 La fonction de transfert $f(e_i)$

La plupart des réseaux utilisés pour les applications prévisionnelles sont organisés en couches, on parle alors de *réseaux à couches* (figure 2). Le transfert des informations s'effectue par propagation, de la couche d'entrée vers la couche de sortie, cela signifie que les neurones d'une couche c reçoivent des informations en provenance de la couche $c-1$, puis renvoient leurs impulsions aux neurones de la couche $c+1$. En terme de connections cela implique qu'une couche c n'est connectée qu'à la couche $c+1$, à cela il faut ajouter que les neurones d'une même couche ne sont pas connectés entre eux.

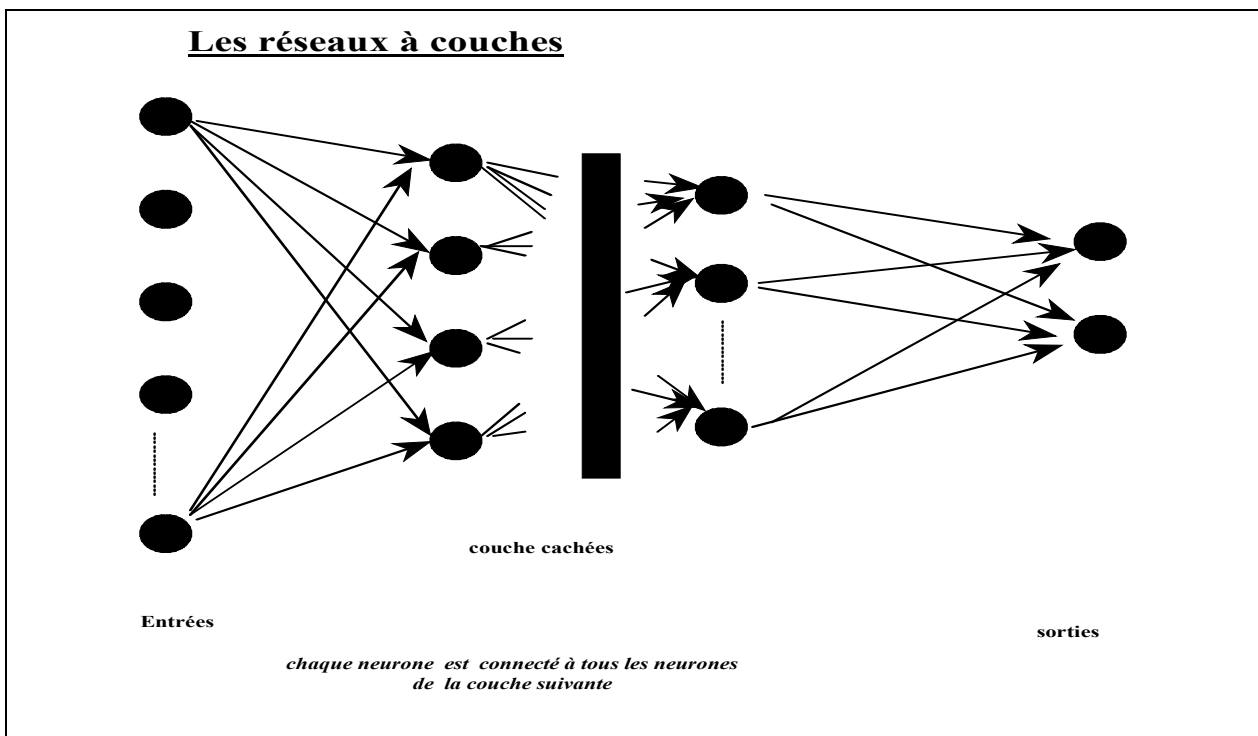


Figure 2

Le calibrage des réseaux s'effectue par l'intermédiaire d'un apprentissage dont l'objectif est de déterminer la juste valeur des connections, de façon à ce que la réponse d'un réseau à une entrée soit la plus proche possible de la sortie réelle. L'apprentissage peut être effectué selon deux modes, supervisé ou non supervisé. L'*apprentissage supervisé* signifie que les connections sont mises à jour à partir d'une fonction coût E à l'issue de la présentation d'une entrée ou d'un ensemble d'entrées. Ce type d'apprentissage est fondé sur la punition lorsque l'erreur est élevée et sur la récompense en cas d'erreur nulle. L'*apprentissage non supervisé* correspond à la situation où le réseau détermine sa propre évolution, on peut parler alors d'auto-organisation du réseau (modèle de Hopfield).

Les premières mises à jours des connections furent réalisées à l'aide de la règle delta dans laquelle les poids étaient incrémentés de ΔW_{ij}

$$\text{où } \Delta W_{ij} = \alpha \cdot \delta_i \cdot S_j = \alpha \cdot (\text{observé}_i - S_i) \cdot S_j$$

Cette règle était limitée aux réseaux à deux couches. Pour les réseaux multicouches il a fallu attendre les travaux de généralisation de cette règle par Le Cun [1985]. Cette généralisation porte le nom de rétropropagation du gradient (ou rétropropagation de l'erreur).

On distingue plusieurs variantes de cette généralisation, les principales sont les suivantes :

- *gradient stochastique* ; il fait intervenir des termes aléatoires au niveau de la fonction de transfert ou encore dans la règle de mise à jour des connections,
- *gradient local* ; on modifie les connexions de façon individuelle pour chaque élément de l'ensemble d'apprentissage,
- *gradient total*, la mise à jour des connexions s'effectue à l'issue de la présentation de tous les individus de l'ensemble d'apprentissage ;
- *gradient conjugué* ; la mise à jour s'effectue en fonction du gradient courant et de celui calculé pour l'individu précédent

Le potentiel d'un réseau de neurones s'évalue à l'aide d'une fonction coût mesurant l'erreur entre la sortie prévue et la sortie réelle. On pourra voir en encadré que l'écriture de la formule de l'erreur n'intervient qu'au niveau des connections de l'avant dernière couche. Cela signifie que l'on est libre de choisir sa propre fonction coût, de façon à maximiser un critère quelconque.

Approche neuronale pour la prévision

Elle consiste à calibrer un réseau heure par heure à partir d'un échantillon contenant des données explicatives et à expliquer, parfois appelées mesures et réponses. Nous effectuons un apprentissage horaire car nous supposons que le trafic est surtout fonction de l'heure c'est à dire

dépendant de la variation horaire (heures creuses par rapport aux heures de pointes) et de la variation calendaire (jours ouvrables par rapport aux jours non ouvrables) déduite des profils.

Cette méthode nécessite trois étapes. La première communément appelée *extraction de données* consiste à sélectionner les données significatives en vue de la phase d'apprentissage. La seconde phase appelée *apprentissage* a pour objectif de trouver la configuration optimale du réseau de façon à ce que l'erreur de prévision soit la plus faible. Cette étape fournira pour chaque heure un jeu de connections associé à un réseau de neurones. La dernière partie appelée *prévision* consiste à prédire le trafic pour un horizon donné à partir d'un réseau de neurones préalablement calibré durant la phase d'apprentissage.

Extraction d'un jeu de données

Cette phase qui permet de constituer le jeu de données en vue de l'apprentissage, consiste à sélectionner des données utiles pour la prévision. En effet au cours de l'apprentissage supervisé, nous devons présenter au réseau un ensemble de couples mesure/réponse dans lequel mesure représente les variables explicatives et réponse la variable à prédire. La plupart du temps ces variables mesures sont issues de relevés effectués par des capteurs à intervalle régulier. On peut aussi se demander quelle est la fréquence à laquelle les mesures doivent être effectuées : l'heure, la demi-heure... ayant à l'esprit qu'une fréquence horaire peut masquer certains problèmes, alors qu'une fréquence élevée peut induire trop de perturbations. Toutefois il est préférable d'avoir trop d'informations que pas assez, puisque il est toujours possible de procéder à un lissage des données de façon à atténuer la variation (par exemple agrégation des débits six minutes en débit horaire).

Pour mieux comprendre quelles données sont extraites en vue de l'apprentissage on pourra se reporter au tableau 1.

Pour améliorer l'estimation il est préférable parfois d'ajouter des variables calculées. Ainsi dans ATHENA [DANECH-ARON] on utilise des variables profils. L'intérêt du profil réside dans le fait qu'il permet d'éliminer l'effet, mettant en évidence une certaine tendance pour un instant donnée.

En résumé pour un horizon de prévision h et selon la notation du tableau 1, on recherche une relation R entre la réponse $q(t+h)$ et la mesure sachant que la mesure est composée de variables brutes $Q(t)$ et de variables calculées $P(t)$. Cette relation peut s'écrire :

$$q(t+h) = R\{Q(t), P(t)\}$$

Cette formalisation nécessite de fixer la longueur du proche passé k pour les données brutes ainsi que m l'étendue du profil. Bien souvent m est supérieur à k , pour la simple raison qu'au delà d'un certain temps ou passé, les données brutes n'apportent plus d'information pour la prévision de trafic. Cependant il en va autrement du profil, qui traduit la variation de trafic sur quelques heures, laquelle a précisément pour vocation de discriminer les individus. Pour que la discrimination soit de bonne qualité il faut nécessairement utiliser des profils de plusieurs heures. Concrètement pour chaque heure on constitue un tableau à partir d'un fichier archive, lequel contient un ensemble de débits et de variables calculées précédant l'instant de prévision.

Tableau 1

exemple des données extraites en vue de la prévision de $q(t+h)$

	données brutes						données calculées			
	$q(t+h)$	$q(t)$	$q(t-1)$	$q(t-2)$	$q(t-k-1)$	$p(t)$	$p(t-1)$..	$p(t-m-1)$
jours	$q_{1,t+h}$	$q_{1,t}$	$q_{1,t-1}$	$q_{1,t-2}$	$q_{1,t-k-1}$	$p_{1,t}$	$p_{1,t-1}$..	$p_{1,t-m-1}$
	$q_{2,t+h}$	$q_{2,t}$	$q_{2,t-1}$	$q_{2,t-2}$	$q_{2,t-k-1}$	$p_{2,t}$	$p_{2,t-1}$..	$p_{2,t-m-1}$

	$q_{j,t+h}$	$q_{j,t}$	$q_{j,t-1}$	$q_{j,t-2}$	$q_{j,t-k-1}$	$p_{j,t}$	$p_{j,t-1}$..	$p_{j,t-m-1}$

	$q_{n,t+h}$	$q_{n,t}$	$q_{n,t-1}$	$q_{n,t-2}$	$q_{n,t-k-1}$	$p_{n,t}$	$p_{n,t-1}$..	$p_{n,t-m-1}$

On note :

- t** l'instant ou l'on effectue la prédiction,
- h** l'horizon de la prédiction,
- per** le nombre de mesures par jour (24 pour un débit horaire, 240 pour un débit six minutes),
- k** la longueur du passé pour les données brutes
- m** l'étendue ou la période couverte par le profil
- d** le nombre de jours
- $q_{j,t}$ le débit du jour j à l'instant t $t \in [1, per]$

Dans le tableau 1 on trouve les données brutes suivantes :

$q(t) = \langle q_{1,t}, q_{2,t}, \dots, q_{j,t}, \dots, q_{d,t} \rangle$, le **vecteur mesure** correspondant aux débits relatifs à l'instant t

$q(t+h) = \langle q_{1,t+h}, q_{2,t+h}, \dots, q_{j,t+h}, \dots, q_{d,t+h} \rangle$, le **vecteur réponse** correspondant aux débits relatifs à l'instant t+h,

$Q(t) = [q(t-k+1) \dots q(t-1) q(t)]$, la **matrice mesure** composée des vecteurs mesures des instants compris entre (t-k-1) et t. On trouve également des

données calculées : $P_{j,t} = \frac{q_{j,t}}{\sum_{i=t-m-1}^t q_{j,i}}$, la **variable profil** du jour j à l'instant t,

$P(t) = \langle P_{1,t}, \dots, P_{2,t}, \dots, P_{d,t} \rangle$, le **vecteur profil** des variables profils relatives à l'instant t,

$P(t) = [P(t), P(t-1), \dots, P(t-m-1)]$, la **matrice profil** des vecteurs profils.

Apprentissage

Il s'effectue en temps différé, on utilise pour cela les données extraites du passé. Pour chaque heure, l'apprentissage nécessite la réalisation des phases successives.

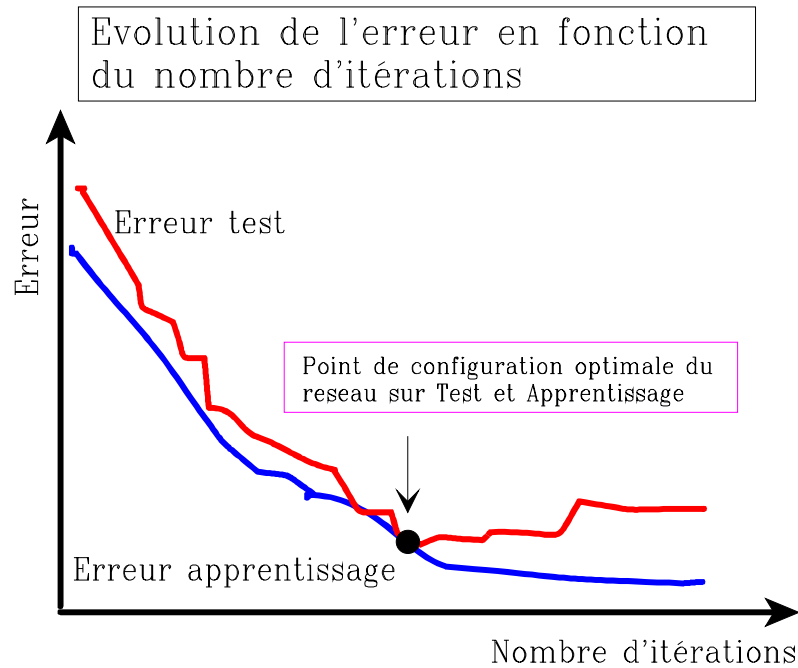


Figure 3

Normalisation des entrées et des sorties

On normalise les données de façon à les ramener sur un même intervalle ici $[0,1]$. On procède pour cela à une translation sur les variables brutes à savoir les débits. Une façon de ramener un débit $q_{i,j} \in [0, \max]$ sur l'intervalle $[0,1]$ consiste à le diviser par Max (Max correspond à la capacité maximum d'un axe autoroutier exprimée par un débit horaire, 5000 véhicules/h par exemple). En ce qui concerne les profils on ne les normalise pas, car par définition ils sont déjà sur l'intervalle $[0,1]$.

La normalisation doit également porter sur la valeur observée relative à l'horizon de prévision. En effet pour être compatible et comparable avec la sortie de la fonction de transfert, celle-ci doit être normaliser de la même manière que les données brutes. La normalisation par Max concerne donc les vecteurs $q(t-k-1)...q(t-1), q(t)..q(t+h)$.

Constitution de l'ensemble test et d'apprentissage

L'apprentissage du réseau nécessite un jeu de données à partir duquel on calibre le réseau; c'est à dire ses connections. Cet ensemble est appelé ensemble d'apprentissage. L'évaluation du réseau s'effectue à partir de données différentes de celles utilisées durant l'apprentissage. Par définition ce dernier jeu de données est appelé ensemble test.

Soit AP l'ensemble d'apprentissage et TEST l'ensemble test, les propriétés ensemblistes sont les suivantes :

$$AP \cap TEST = \emptyset \quad AP \cup TEST = \Omega$$

La constitution de ces ensembles peut être effectuée de façon aléatoire ou arbitraire, ceci en fonction de la taille du jeu de données disponibles.

Recherche du réseau donnant une erreur de prévision minimale

Une fois la normalisation et la constitution des ensembles test et d'apprentissage réalisées on peut lancer la phase d'apprentissage. Le processus d'apprentissage consiste à initialiser les connexions de façon aléatoire puis à présenter successivement les individus de l'ensemble d'apprentissage de façon à faire décroître l'erreur individuelle. Les connexions sont mises à jour par un gradient locale ce qui évite de rester bloquer sur un minimum local. Le réseau apprend jusqu'à ce qu'il atteigne en un certain nombre d'itérations un optimum sur les deux ensembles d'apprentissage et de test. Cet optimum est fonction de l'initialisation mais aussi de la topologie choisie, ce qui signifie que la sélection de l'optimum peut nécessiter plusieurs tentatives, à la fois pour cerner la bonne topologie et la bonne initialisation.

Mémorisation des paramètres,

L'optimum atteint précédemment se traduit par un ensemble de valeurs pour les connexions. Ces connexions ne sont valables que pour un horizon de prévision égal à $t+h$, aussi afin de pouvoir réutiliser le réseau ultérieurement on a besoin de les mémoriser.

On recherche essentiellement les valeurs des connexions correspondant à un optimum obtenu sur les ensembles d'apprentissage et de test. En effet, la principale qualité d'un réseau est de bien s'adapter à l'ensemble d'apprentissage, ce qui se traduit par une décroissance de l'erreur sur l'ensemble d'apprentissage lorsque le nombre d'itérations croît. L'erreur calculée sur l'ensemble test décroît pour sa part jusqu'en un certain nombre d'itérations, mais on assiste delà à une remontée de l'erreur (voir figure 3)

Or l'ajustement obtenu par un apprentissage maximum n'est pas toujours satisfaisant au regard de la population réelle. Pour tenter de déterminer au mieux un ajustement il faut procéder après chaque cycle d'apprentissage à une évaluation du réseau de neurones à l'aide de l'ensemble test de façon à garder l'optimum atteint sur ces deux ensembles. L'optimum ainsi sélectionné, se rapprochera plus facilement de celui que l'on recherche pour la population réelle.

Cette phase détermine les connexions de façon à ce que les sorties prévisionnelles du réseau correspondent au mieux aux valeurs réelles. Pour cela on utilise une fonction coût, généralement quadratique, qui mesure un écart entre la valeur prévue et observée. La mise à jour des connexions s'effectue à l'aide d'une rétropropagation du gradient de l'erreur (voir encadré).

Prévision

Cette phase, qui s'effectue en temps réel est nettement plus simple et plus courte que l'étape d'apprentissage. Lorsque l'on souhaite prévoir un débit associé à l'instant $t+h$, on réalise pour cela les opérations suivantes :

Chargement des paramètres du réseau retenu

On récupère les valeurs des connexions déduites de l'étape de l'apprentissage pour la tranche horaire h , ainsi que la longueur du passé k et la période du profil m .

Constitution des entrées du réseau

On extrait de la base courante les données nécessaires pour constituer les données brutes et pour calculer les données profils de la période **m**. Il reste ensuite à normaliser les données brutes.

Calcul de la sortie prévisionnelle du réseau

Si pour une observation **t**, on souhaite prévoir le débit à **t+h**, cela suppose de présenter en entrée les données constituées précédemment, puis propager les sorties des neurones d'une couche vers les neurones de la couche supérieure, jusqu'au neurone de la dernière couche. Celui-ci fournit la valeur prévisionnelle qui appartient à l'intervalle $[0,1]$. Pour que cette dernière soit compatible avec un débit, on la dénormalise en la multipliant par **Max**.

Le calcul de la prévision s'effectue selon la séquence suivante : soit **x** le vecteur mesure présenté en entrée, **x** s'écrit :

$x = \langle x_1, x_2 \dots x_l, x_{l+1} \dots x_n \rangle$ avec $x_1, x_2 \dots x_l$ les variables brutes normalisées et $x_{l+1} \dots x_n$ les variables profils.

On calcule la sortie des neurones d'entrée de la couche **c-1** :

$$\forall i = 1, n \quad S_i^{c-1} = x_i$$

On calcule la sortie des neurones de la couche cachée **c**

$$\forall i = 1, n \quad S_i^c = f\left[\sum_{j=1}^{nc-1} W_{ij}^{c-1} \cdot S_j^{c-1}\right]$$

On calcule la sortie du neurone **i** de la couche de sortie **c+1**

$$S_i^{c+1} = f\left[\sum_{j=1}^{nc} W_{ij}^c \cdot S_j^c\right]$$

On dénormalise la sortie du réseau de façon obtenir un débit prévisionnel pour l'instant **t+h**

$$q(t+h) = S_i^{c+1} \cdot \text{Max}$$

APPLICATIONS

On a utilisé des mesures demi-horaires relevées en milieu autoroutier (voir figure 4). Il s'agissait de prévoir le débit du poste de comptage de Beaune dans le sens Paris-Provence, pour un horizon d'une heure. Les données portent sur les postes de comptage suivants :

Beaune (autoroute A6 en provenance de Paris, pk 305.3)

Avallon (autoroute A6 en provenance de Paris, pk 211.8)

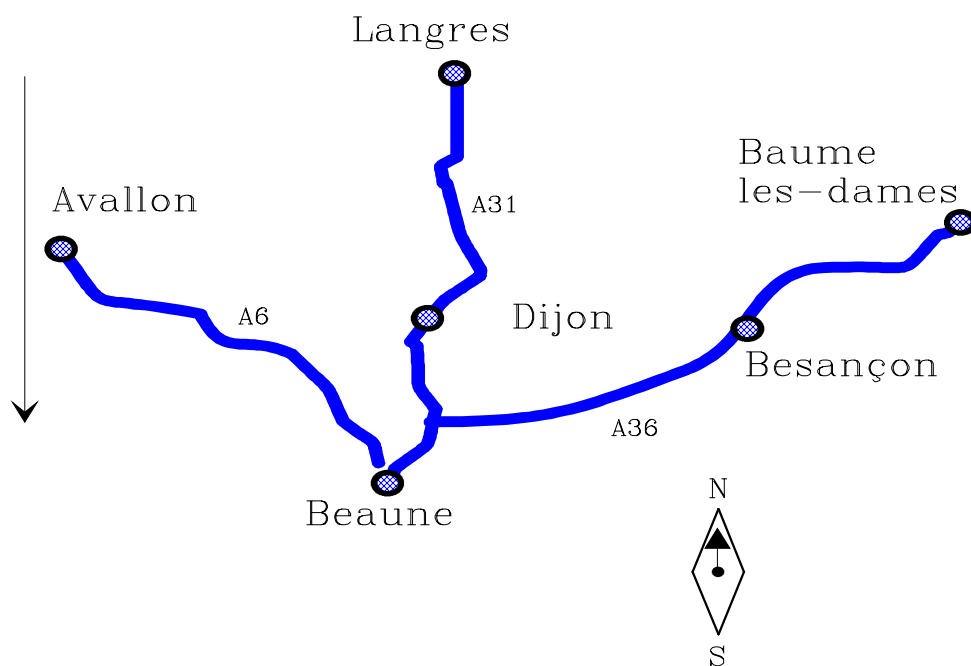
Baume les dames (autoroute A36 en provenance de Mulhouse, pk 91.0)

Langres (autoroute A31 en provenance de Nancy, pk 98.8)

Beaune est le point sur lequel porte la prévision, alors que Avallon, Baume les dames et Langres sont des postes amonts dont les données sont riches d'information pour la prévision. Pour l'apprentissages on a utilisé les données enregistrées durant les mois de Juillet et Août des années 1984 à 1989, dont on a écarté les jours aberrants (trop de données nulles ou incertaines). Les mesures des mois de Juillet et Août 1990 constitueront l'ensemble test.

Disposant de données demi-horaires, on aurait pu envisager de calibrer quarante-huit réseaux de neurones. Cependant on admet que la variation d'une demi-heure à l'autre n'est pas significative. On conserve donc le même réseau pour l'heure entière et la demi-heure suivante. Ceci a pour conséquence de doubler les effectifs des ensembles d'apprentissage et test ce qui ne peut que rendre plus robuste les résultats.

Beaune sens Paris–Province



Le réseau choisi

C'est un réseau à trois couches muni d'une couche d'entrée ou rétine, d'une couche cachée (composée de 5 neurones) et d'une couche de sortie (figure 5). Le nombre de neurones sur la couche d'entrée est fonction du proche passé k des données brutes et de la période m du profil.

Pour quatre postes de comptage, $n=4.k+m$

Pour mettre à jour les connections, nous utilisons un gradient local, dont on trouve la description détaillée en encadré. Ce type d'apprentissage a l'avantage d'individualiser la mise à jour des connections. Cependant nous aurions pu adopter un gradient total, qui aurait sans doute permis d'accélérer la vitesse de calibrage du réseau. Le choix de la règle de mise à jour constitue une perspective d'amélioration sachant qu'il existe des techniques d'accélération pour le gradient [FAHLMAN, 1989]. Etant donné que le réseau ne comporte qu'un neurone de sortie, cela induit une simplification de la mise à jour des connections de la couche d'entrée vers la couche cachée.

Nous avons choisi une valeur faible pour α (constante de pondération du gradient), nous l'avons fixé à 0.01. Il est cependant envisageable de donner à α une valeur élevée au début de l'apprentissage, accélérant par voie de conséquence la convergence du réseau, puis de faire décroître cette valeur avec le nombre d'itérations ce qui modifie de façon infime les connexions. augmente, α décroît ce qui modifie de façon infime les connections.

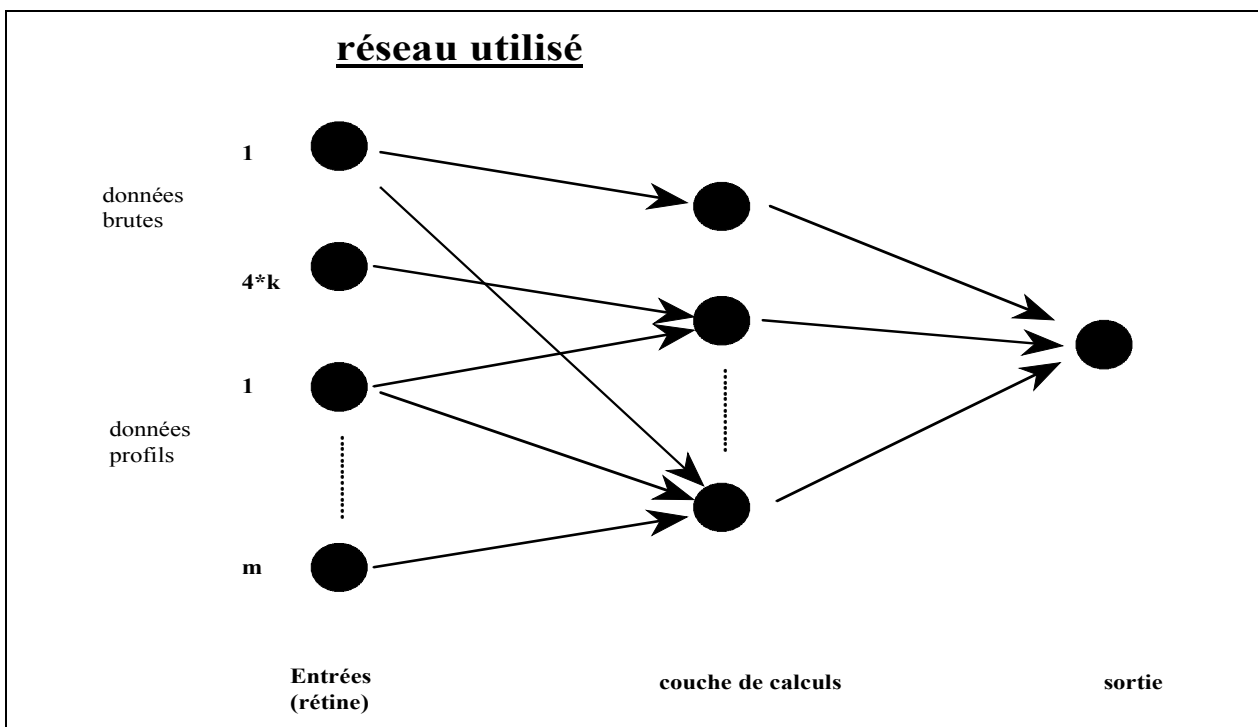


Figure 4 le réseau choisi

La fonction de transfert est une sigmoïde $f(x) = \frac{1}{1 + e^{-x}}$ dont la dérivée est $f'(x) = f(x).(1 - f(x))$

On utilise en entrée les grandeurs ci-dessous normalisées :

débits du point **QP(t) ...QP(t-k-1)**
 débits amont 1 **QA₁(t)...QA₁(t-k-1)**
 débits amont 2 **QA₂(t)...QA₂(t-k-1)**
 débits amont 3 **QA₃(t)...QA₃(t-k-1)**
 variables calculées (profil) **Z₁...Z_m**

et l'on cherche en sortie la grandeur QP(t+1) dénormalisée pour être compatible avec le débit prédit. Les variables **Z₁ à Z_m** sont calculées pour chaque instant de la prédiction (rappelons que la période **m** est bien souvent plus longue que **k** le proche passé des données brutes).

En résumé le réseau de neurones cherche une approximation d'une fonctionnelle \mathbb{R} de sorte que la prévision soit la plus proche possible du débit réel.

$$QP(t+1) = \text{Max}.R\left\{\frac{QP(t-s)}{MPax}, \frac{QA_i(t-s)}{Max}, Z_1, Z_2 \dots Z_m\right\}$$

Pour $i \in [1, 3]$, $s \in [0, k-1]$, Max étant la constante de normalisation.

Dans cette application on dispose de stations situées en amont de Beaune, ce qui permet d'améliorer la prévision car les véhicules passant par ces amonts arriveront à Beaune environ une heure plus tard. De plus en fonction du type de circulation fluide ou congestionné, ce ne sont pas les mêmes mesures au niveau des amonts qui interviennent dans la prédiction.

Ainsi en situation fluide, il est raisonnable de penser que les mesures de l'heure précédente interviennent dans la prévision, tandis qu'en situation congestionné il faut remonter à plus d'une heure. On peut penser que le réseau utilise l'information profil du point pour discriminer les types de situation de trafic.

Résultats

Pour évaluer la méthode décrite, nous utilisons deux critères d'erreur. Le premier indiquant un taux global d'erreur quadratique moyenne, c'est calculé à l'aide de la formule suivante :

$$\sqrt{\frac{\sum_{i=1}^n (prévu_i - observé_i)^2}{\sum_{i=1}^n (observé_i)^2}}$$

Le second, donne une erreur globale en terme de nombre de véhicules, est calculé par :

$$\sqrt{\frac{\sum_{i=1}^n (prévu_i - observé_i)^2}{n}}$$

Signalons que d'autres utilisent des formules différentes, ce qui rend délicate la comparaison.

Données horaires

Pour notre première expérience nous avons agrégé les données de façon à constituer des données horaires (profils et débits horaires). Les ensembles nécessaires aux calibrages des réseaux comprenaient respectivement pour l'apprentissage les mois de juillet et août des cinq années 1984 à 1988 pour le test les mois de juillet et août 1989.

L'horizon de prévision a été fixé à une heure ($h=1$), tandis que la période de profil était de six heures ($m=6$). Nous avons fait varier la longueur du proche passé pour les données brutes d'une à quatre heures ($k \in [1, 4]$) et examiné l'influence de sa variation à m fixé.

Evoquant précédemment le problème de la topologie, nous avons laissé entendre que les résultats étaient fonction des paramètres h, m, k mais aussi de la topologie du réseau de neurones. Ceci est le second élément que nous avons fait varier, en utilisant pour la prévision plusieurs types de réseaux différant par le nombre de neurones sur la couche cachée. Nous avons testé les typologies M1, M2, M3 avec respectivement cinq, quatre et trois neurones sur la couche cachée. par conséquent en plus de la variation de k ($k \in [1..4]$), on utilisera pour la prévision.

Le résultat de ces expériences sont présentés dans le tableau 2. On remarque que le meilleur résultat est obtenu avec la typologie M1, celle disposant du plus grand nombre de neurones sur la couche cachée. Globalement la topologie M1 donne de meilleurs résultats au niveau de l'ensemble test pour k appartenant à 1 à 3. On ne peut pas en dire autant pour l'ensemble d'apprentissage, mais cela n'est pas significatif car il est très difficile de départager des réseaux à partir d'un ensemble d'apprentissage en raison même de leur capacité d'apprentissage.

Pour $k=2$ (un proche passé de deux heures) on obtient un taux d'erreur global de 7% pour l'apprentissage et 8,78% pour l'ensemble test (tableau 2). Nous en déduisons que deux heures suffisent pour bien prédire le débit de Beaune.

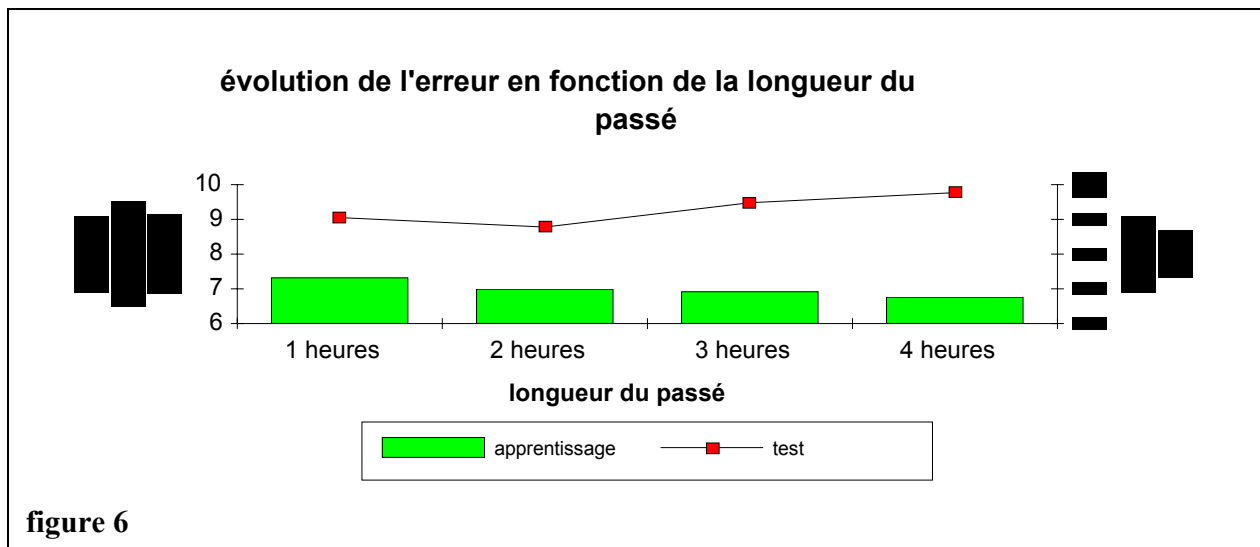
Erreur globale pour un profil de 6 six heures				
proche passé de 1 heure				
	M1	M2	M3	
Apprentissage	7,32	7,21	7,63	
Test	9,05	9,06	10,08	
proche passé de 2 heures				
	M1	M2	M3	
Apprentissage	6,99	6,89	6,9	
Test	8,78	9,2	9,2	
proche passé de 3 heures				
	M1	M2	M3	
Apprentissage	6,92	6,83	6,8	
Test	9,48	9,71	9,8	
proche passé de 4 heures				
	M1	M2	M3	
Apprentissage	6,76	6,69	6,87	
Test	9,77	9,42	9,39	

M1 5 neurones couche cachée
M2 4 neurones couche cachée
M3 3 neurones couche cachée

Tableau 2

L'influence de la topologie semble minime comme le montre les résultats. L'apprentissage réalisé sur la topologie M2 est de meilleure qualité que sur les autres topologies. Cependant les résultats sur les ensembles tests sont meilleurs pour la topologie M1. C'est pourquoi on accorde une plus grande confiance à cette dernière.

La figure 6 retrace les résultats globaux obtenus par la topologie M1. On constate que l'augmentation de la longueur du passé se traduit par amélioration de l'erreur sur l'ensemble d'apprentissage, alors qu'au niveau de l'ensemble test on note un gain jusqu'à deux heures, tandis qu'au delà l'erreur remonte. Ceci correspond à un phénomène de saturation, de la part du réseau qui n'arrive plus à discriminer l'information utile.



Données demi-horaires

Pour la dernière expérience l'ensemble d'apprentissage contenait les mois de Juillet et d'Août des années 1984 à 1989, tandis que l'ensemble test contenait Juillet et Août 1990. Ici la prévision contrairement à l'expérience précédente a été réalisée avec les données demi-horaires et des profils demi-horaires, les paramètres m , k valant 8, et 2. Cela représente un profil de 4 heures et un passé de deux demi-heures ($Q(h)$ et $Q(h-1/2)$). Nous espérons que le passage d'un modèle horaire à un modèle demi-horaire pourrait améliorer les résultats.

A titre de référence nous donnons les résultats obtenus par ATHENA lors de son application en Juillet et Août 1990 sur la station de comptage de Beaune. Bien que l'on ne puisse complètement comparer notre démarche neuronale et ATHENA. En effet le calibrage d'ATHENA est indépendant de l'ensemble test, alors que le calibrage d'un réseau de neurones nécessite un ensemble d'apprentissage et un ensemble test pour fixer l'optimum des connections. Cependant les résultats obtenus sur les ensembles d'apprentissage et de test à savoir 8,5% et 8,2% tendent à montrer que la méthode semble robuste et permettent d'envisager une véritable comparaison dans le future.

La figure 7 compare les taux d'erreur moyenne par mois sur les prédictions obtenues par la méthode neuronale et par ATHENA, tans que la figure 8 donne les erreurs en terme de nombre de véhicules.

Erreur globale pour la prévision de Beaune sens Paris-Provence

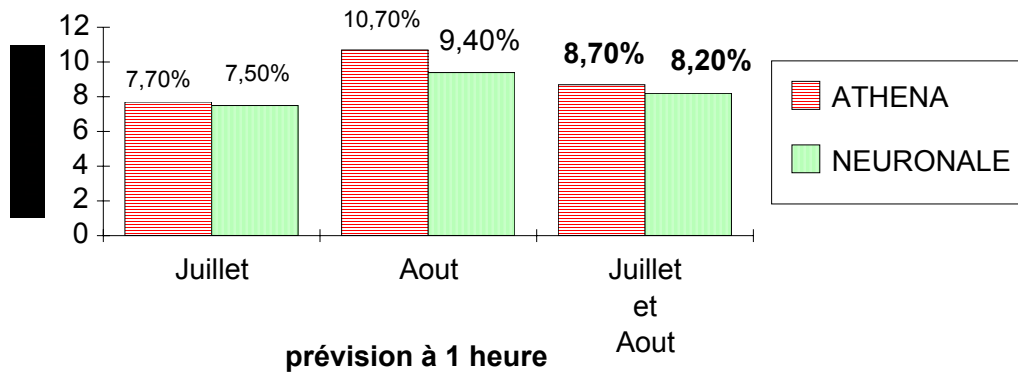


figure 7

Erreur globale pour la prévision de Beaune sens Paris-Provence

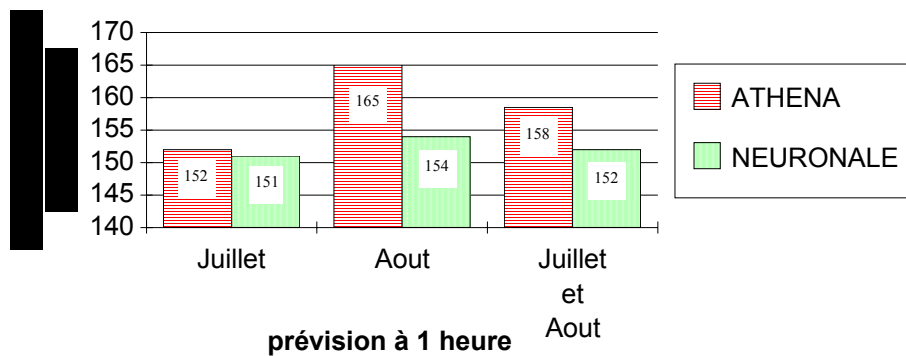


figure 8

Perspectives et conclusion

Les perspectives d'amélioration du modèle prévisionnel par réseaux de neurones résident soit dans une réduction du nombre de réseaux, soit dans les techniques d'accélération du gradient. On peut envisager par exemple de procéder à une classification des profils, de façon à effectuer un travail discriminant avant d'introduire en entrée d'un réseau généralisé le résultat de cette classification, ainsi que les mesures et les profils. Pour l'utilisation temps réel cela suppose de

reconnaître la classe d'appartenance du profil avant de propager le vecteur d'entrée à travers le réseau.

Cependant il est déjà possible d'améliorer notre modèle en conservant les sorties des prévisions antérieures à t , que l'on introduirait dans le réseau pour la prévision à l'instant t du trafic $t+h$. Ceci aurait pour avantage de constituer un réseau avec une mémoire qui pourrait concentrer l'information. Grâce à ce type de réseau il est possible d'introduire les erreurs prévisionnelles de manière à en tenir compte pour les prévisions futures. Ce type de réseau étant largement utilisé pour le traitement de la parole dans les réseaux de neurones avec retard temporel [Waibel, 1989].

Pour conclure, on peut dire que les expériences nous ont permis de montrer l'intérêt que peut présenter une approche neuronale pour une application prévisionnelle. L'avantage de cette démarche réside dans le fait que l'on simplifie considérablement le calibrage qui se résume simplement en une phase d'apprentissage d'un réseau de neurones. De plus la prévision est très simple puisqu'il ne s'agit que d'une propagation d'un vecteur de profils et de débits.

Globalement cette méthode permet d'effectuer de bonnes prévisions puisque l'erreur obtenue sur l'ensemble test est de 8,2% ce qui répond au critère de qualité que nous nous étions fixé. De plus, sachant que l'analyse a porté sur les jours les plus chargés de l'année, on peut attendre de la méthode un comportement encore meilleur pour les autres jours de l'année.

BIBLIOGRAPHIE

Bourret P., Reggia J., Samuelides M. – Réseau Neuronaux, Tekena éd., 1991.

Box G. et Jenkins G. - Time Series Analysis, Forecasting and Control, Holden Day ed., 1970

Danech-Pajouh M., Aron M., ATHENA, une méthode de prévision à court terme du trafic autoroutier, Recherche Transports et Sécurité N°28, décembre 1990.

Dougherty M., Analysis of the Utrecht Traffic Data Using Neural Network, Technical Note 344, May 1993.

Le Cun Y., une procédure d'apprentissage pour réseau à seuil asymétrique, Proc. of Cognitive Cesta-Afcet 1985.

Le Monde, Neurones : la lyonnaise se jette à l'eau", Le Monde Informatique 4 Novembre 1991

Mangeas M., Advantages of Multilayer Perceptron for Modelling and Forecasting Time Series : Application To The Daily Electrical Consumption In France, Proceedings de Neuro-Nimes 1993.

Scott E., Fahlman ---Faster Learning Variations of Backpropagation : "An empirical Study Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufman ed., 1989.

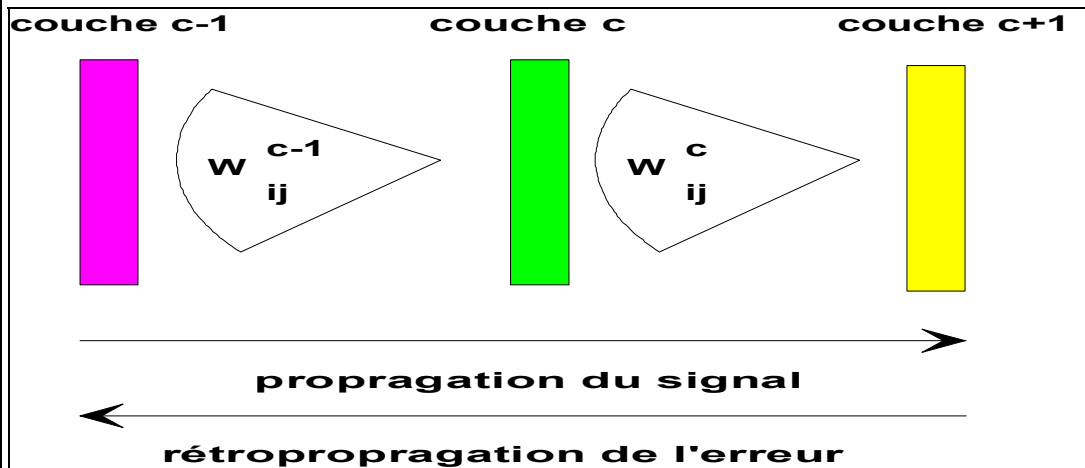
Waibel A. Phoneme Recognition Using Time-Delay Neural Network, IEEE Transactions on Acoustics, Vol 37 N° 3, May 1989

Encadré

Règle de mise à jour des connections à l'aide d'un gradient local

Dans la phase de rétropropagation la modification des poids se fait de la façon suivante :

$$W_{ij}^c = W_{ij}^c + \alpha \frac{\partial E}{\partial W_{ij}^c} \quad (1)$$



$$E = (S_i^{c+1} - \text{Observé}_i)^2 \quad (2)$$

La rétropropagation du gradient se calcule différemment selon que l'on modifie les poids reliés à la dernière couche ou bien les poids des couches précédentes. On parle de rétropropagation car la mise à jour des connexions d'une couche utilise une partie du gradient de l'erreur des couches précédentes.

Pour un neurone i appartenant à la dernière couche :

La fonction de coût E se compose de plusieurs fonctions (f,S,e). Aussi pour exprimer

correctement la gradient décompose-t-on tout d'abord $\frac{\partial E}{\partial W_{ij}^c}$ à l'aide de e_i^{c+1} entrée avec W_{ij}^c .

$$\frac{\partial E}{\partial W_{ij}^c} = \frac{\partial E}{\partial e_i^{c+1}} \cdot \frac{\partial e_i^{c+1}}{\partial W_{ij}^c} = \frac{\partial E}{\partial e_i^{c+1}} \cdot \frac{\partial \sum_{l=1}^{nc+1} W_{il}^c \cdot S_l^c}{\partial W_{ij}^c} \quad (3)$$

La décomposition de (3) s'écrit :

$$\frac{\partial E}{\partial W_{ij}^c} = \frac{\partial E}{\partial e_i^{c+1}} \cdot S_j^c \quad (4)$$

Il faut à nouveau décomposer $\frac{\partial E}{\partial e_i^{c+1}}$ car E est fonction de S_i^{c+1} , lui-même issu de e_i^{c+1}

$$\frac{\partial E}{\partial e_i^{c+1}} = \frac{\partial E}{\partial e_i^{c+1}} \cdot \frac{\partial S_i^{c+1}}{\partial e_i^{c+1}} = (\text{observé}_i - S_i^{c+1}) \cdot f'(e_i^{c+1}) \quad (5)$$

La règle de mise à jour des connexions est donc finalement :

$$W_{ij}^c = W_{ij}^c + \alpha (\text{observé}_i - S_i^{c+1}) \cdot f'(e_i^{c+1}) S_j^c$$

Pour un neurone i appartenant aux autres couches :

Comme dans l'équation (3), on décompose le gradient $\frac{\partial E}{\partial W_{ij}^c}$ à l'aide de l'entrée e_i^c à partir de W_{ij}^{c-1} .

$$\frac{\partial E}{\partial W_{ij}^{c-1}} = \frac{\partial E}{\partial e_i^c} \cdot \frac{\partial e_i^c}{\partial W_{ij}^{c-1}} = \frac{\partial E}{\partial e_i^c} \cdot S_j^{c-1} \quad (6)$$

L'entrée e_i^c est utilisée au niveau des entrées des couches supérieures, de sorte que l'on peut

écrire $\frac{\partial E}{\partial e_i^c}$ à l'aide des gradients des entrées $\frac{\partial E}{\partial e_i^{c+1}}$ déjà calculé en (5).

$$\frac{\partial E}{\partial e_i^c} = \sum_{l=1}^{nc+1} \frac{\partial E}{\partial e_l^{c+1}} \cdot \frac{\partial e_l^{c+1}}{\partial e_i^c} \quad (7)$$

On décompose la gradient $\frac{\partial e_l^{c+1}}{\partial e_i^c}$ à partir de la sortie S_i^c intervenant dans le calcul de e_i^{c+1} :

$$\frac{\partial e_l^{c+1}}{\partial e_i^c} = \frac{\partial e_l^{c+1}}{\partial S_i^c} \cdot \frac{\partial S_i^c}{\partial e_i^c} = W_{li}^c \cdot f'(e_i^c) \quad (8)$$

$$\text{Car } \frac{\partial e_l^{c+1}}{\partial S_i^c} = \frac{\partial \sum_{j=1}^{nc} W_{lj}^c S_j^c}{\partial S_i^c} = W_{lj}^c \text{ et } \frac{\partial S_i^c}{\partial e_i^c} = \frac{\partial f(e_i^c)}{\partial e_i^c}$$

On rapporte le gradient $\frac{\partial e_i^{c+1}}{\partial e_i^c}$ dans (7) :

$$\frac{\partial E}{\partial e_i^c} = \sum_{l=1}^{nv+1} \frac{\partial E}{\partial e_l^{c+1}} W_{li}^c \cdot f'(e_i^c) \quad (9)$$

Finalement le gradient $\frac{\partial E}{\partial W_{ij}^{c-1}}$ s'écrit :

$$\frac{\partial E}{\partial W_{ij}^{c-1}} = \sum_{l=1}^{nc+1} \left(\frac{\partial E}{\partial e_l^{c+1}} \cdot W_{li}^c \right) \cdot f'(e_i^c) \cdot S_j^{c-1} \quad (10)$$

La règle de mise à jour des connexions des neurones des autres couches est donc :

$$W_{ij}^{c-1} = W_{ij}^{c-1} + \alpha \sum_{l=1}^{nc+1} \left(\frac{\partial E}{\partial e_l^{c+1}} \cdot W_{li}^c \right) \cdot f'(e_i^c) \cdot S_j^{c-1}$$

Résumé

Cet article présente les résultats de travaux visant à introduire une approche neuronale dans une méthode de prévision de trafic à court terme. Cette approche repose sur l'hypothèse d'une forte dépendance entre l'heure et le trafic autoroutier. Partant de cette hypothèse, on découpe l'échantillon historique de façon à attribuer à chaque réseau une tranche horaire particulière. Les paramètres du réseau sont ensuite estimés par le biais d'un apprentissage supervisé. Grâce au modèle neuronal la prévision n'est qu'une propagation d'un signal d'entrée (débits, profils) au travers d'un réseau de neurones. Cette méthode est évaluée à l'aide de mesures autoroutières relevées durant les mois de juillet et d'août de plusieurs années. Sur ces données, on atteint une erreur globale satisfaisante, ce qui ouvre comme perspective de recherche la généralisation du modèle.

Abstract

This paper presents the results of research aimed at applying a neural approach to short term traffic forecasting. This work is based on the hypothesis that there is a strong link between time of day and levels of motorway traffic. On this basis the historical sample was split up so as to assign a specific period of time to each network. Values for network parameters were then found by means of supervised training. With the neural model forecasting is merely a question of propagating in input signal (flows, profiles) through a neural network. This approach was evaluated by using motorway measurements gathered during July and August over several years. The results provided a level of general error which was satisfactory and open up possibilities for future research, for example the more general application of this model.